Graph Invariant Kernels*

Francesco Orsini[1,2] Paolo Frasconi[2]
Luc De Raedt[1]

[1]Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3001 Heverlee, Belgium
{francesco.orsini,luc.deraedt}@cs.kuleuven.be

[2]Department of Information Engineering
Università degli Studi di Firenze
Via di Santa Marta 3
I-50139 Firenze, Italy
paolo.frasconi@unifi.it

Abstract

We introduce a novel kernel that upgrades the Weisfeiler-Lehman and other graph kernels to effectively exploit high-dimensional and continuous vertex attributes.

Graphs are first decomposed into subgraphs. Vertices of the subgraphs are then compared by a kernel that combines the similarity of their labels and the similarity of their structural role, using a suitable vertex invariant. By changing this invariant we obtain a family of graph kernels which includes generalizations of Weisfeiler-Lehman, NSPDK*, and propagation kernels.*

We demonstrate empirically that these kernels obtain state-of-the-art results on relational data sets.

1. Introduction

Renewed interest has been manifested in recent years on graph kernels which can handle continuous (possibly high dimensional) attributes. A vast body of literature on Graph kernels is devoted to symbolic only structure. Which means that vertices (and possibly edges) are labeled by a number of discrete attributes. Graphs with continuous attributes have been much less investigated. In our work [6] we upgrade existing graph kernels to continuous attributes by using graph and vertex invariants. Vertex invariants are functions that color vertices of a graph in a way that is not affected by isomorphism. They form the basis for several practical isomorphism checking algorithms [4]. We consider the commonalities between graph kernels like the Weisfeiler-Lehman graph kernel (WLGK) [7], the neighborhood subgraph pairwise distance kernel (NSPDK) [1], the propagation kernels [5] and GRAPHHOPPER [2] and summarize them in a general formulation which we call Graph Invariant Kernels (GIK, pronounce "Geek") [6]. GIKs decompose graphs into sets of vertices which are compared by a kernel that measures both their attribute and their structural similarity. The structural similarity indicates to which extent vertices play the same role in the graph they belong to. Our formulation allows arbitrary patterns (e.g. other than the shortest paths used by GRAPHHOPPER) and arbitrary graph and vertex invariants that can be obtained with color propagation schemas (e.g. Weisfeiler-Lehman, Propagation kernel). We also propose spectral coloring which exploits eigen-decompositions.

We show that upgrading graph kernels to continuous values provided by GIKs perform very well on a number of new and existing benchmarks. We experiment with different types of vertex invariants, including Weisfeiler-Lehman and spectral colors and compare the shortest paths used by GRAPHHOPPER with neighborhood subgraphs.

2. Graph Invariant Kernels

GIKs measure the similarity between two attributed graphs G and G' by comparing their vertices with a suitable kernel function $k_{\text{ATTR}}(v, v')$ between the continuous attributes ℓ_c and reweighting their similarity with a function $w(v, v')$:

$$k(G, G') = \sum_{v \in V(G)} \sum_{v' \in V(G')} w(v, v') k_{\text{ATTR}}(v, v'). \quad (1)$$

We define $w(v, v')$ as a count on common graph invariants:

$$w(v, v') = \sum_{\substack{g \in \mathcal{R}^{-1}(G) \\ g' \in \mathcal{R}^{-1}(G')}} k_{\text{INV}}(v, v') \frac{\delta_m(g, g')}{|V_g||V_{g'}|} \mathbf{1}\{v \in V_g \wedge v' \in V_{g'}\}. \quad (2)$$

The weight $w(v, v')$ measures the structural similarity between vertices and can be designed combining an $\mathcal{R}$-decomposition relation [3], a function $\delta_m(g, g')$ and a kernel on vertex invariants k_{INV}.

An $\mathcal{R}$-decomposition relation is a binary relation $\mathcal{R}(G, g)$ that encodes that "g is part of G" and specifies

*A long version of this paper is to appear at IJCAI 2015 a decomposition of G into its *parts* (*patterns*). We denote with $\mathcal{R}^{-1}(G)$ the multiset of all patterns in G.

According to Equation 2 the weight between a pair of vertices increases whenever the two vertices appear in the same pattern with the same structural role.

The function $\delta_m(g, g')$ is used to determine whether the two patterns match, while the indicator function $\mathbf{1}\{v \in V_g \wedge v' \in V_{g'}\}$ is introduced to select only the subgraphs g and g' in which the vertices v and v' are involved respectively.

The kernel function k_{INV} is used to measure the similarity between the colors produced by a vertex invariant $\mathcal{L}$ and encodes the extent to which the vertices play the same structural role in the pattern. A complete vertex invariant gives the most fine-grained matches and has the same effect as using an isomorphism map f, while weaker invariants induce spurious matches. Weaker invariants can be desirable because they allow to compare non isomorphic graphs.

3. Contribution

In our work [6] we use GIKs to naturally upgrade some existing graph kernels (i.e. Weisfeiler-Lehman Graph Kernels, propagation kernels and NSK) to continuous attributes, we also propose spectral coloring to define vertex invariants.

With spectral coloring we introduce for the first time spectral methods for graph matching in the context of graph kernels. The spectral graph kernel is a novel kernel because exploits spectral coloring.

In [6] we propose graph datasets with continuous attributes. SYNTHETIC$_{\text{NEW}}$ and FRANKENSTEIN are synthetic, ENZYMES$_{\text{SYMM}}$ and PROTEIN are about bioinformatics. QC and WEASEL are about natural language processing, question classification and hedge cue detection respectively.

In [6] we show the results of our experiments and prove that we obtain the state of the art on five out of six datasets, with our method. We verify that in our experiments it is always the case that continuous attributes increase the accuracy.

To understand the benefits of graphs with continuous attributes in the context of natural language processing we experiment on two graph representations of a sentence. One using words as discrete symbols and one using 300-dimensional word vectors. We show that not only using 300-dimensional word vectors is beneficial, but also that GIKs can better exploit this representation compared to GRAPHHOPPER.

4. Conclusion

The GIK reformulation of well known graph kernels allows to obtain more insights in the exploration of graph kernels with continuous attributes. The underlying idea was to employ vertex invariants for soft subgraph matching. We contributed new insights into graph kernels and to upgrade existing ones for use with continuous attributes. Several graph-kernel instances were then empirically evaluated on a number of new and existing benchmark datasets. The results showed that some combinations of graph and vertex invariants with continuous attributes lead to excellent performance. For future work we plan to improve runtime and scalability focusing on the local GIKs and spectral coloring which are more appealing for sketching.

References

[1] F. Costa and K. De Grave. Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 26th International Conference on Machine Learning*, pages 255–262, 2010.

[2] A. Feragen, N. Kasenburg, J. Petersen, M. de Bruijne, and K. Borgwardt. Scalable kernels for graphs with continuous attributes (see also erratum). In *Advances in NIPS*, pages 216–224, 2013.

[3] D. Haussler. Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.

[4] B. D. McKay and A. Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94–112, 2014.

[5] M. Neumann, N. Patricia, R. Garnett, and K. Kersting. Efficient graph kernels by randomization. In *Machine Learning and Knowledge Discovery in Databases*, pages 378–393. Springer, 2012.

[6] F. Orsini, P. Frasconi, and L. De Raedt. Graph invariant kernels. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. IJCAI, 2015.

[7] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *The Journal of Machine Learning Research*, 12:2539–2561, 2011.